International Journal of Computer Science Engineering and
Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 5, Issue 1, Feb 2014, 127-134

CONGESTION CONTROL BY MODIFIED TCP TAHOE FOR WCDMA UMTS NETWORK

NEETU SHARMA[1] & VIJAY SINGH RATHORE[2]

[1]Department of Computer Engineering, Gyanvihar University, Jaipur, Rajasthan, India

[2]Director, Shri Karni Group of Institutions, Jaipur, Rajasthan, India

ABSTRACT

Third generation of GSM technology (3GSM) has a Wideband-CDMA (W-CDMA) air interface. W-CDMA includes a shared high-speed channel for traffic from the base station to the mobile users. This research paper performs some modification in the Transmission Control Protocol's (TCP) Tahoe variant in UMTS Network. Modifications in TCP Tahoe perform by change the algorithm of the TCP Tahoe; it can perform better to wireless links, and also maintain its uses on the wired networks. This is very desirable feature because the conventional TCP in most cases implement for wired networks and the demand of wired networks is differ the wireless links of the network.

KEYWORDS: Universal Mobile Telecommunication System (UMTS), Transmission Control Protocol Tahoe (TCP Tahoe), TCP RENO, TCP NEW RENO, TCP SACK. & Operational Network Evaluation Tool (OPNET)

INTRODUCTION

TCP is designed for wired networks so it is not perform well in wireless networks. TCP is reliable protocol and any data loss in transmission is due to congestion occurring in the network rather than an unreliable medium as in wireless network. Wireless links are unreliable and packet loss perform all the time due to interference, channel fading, higher bit error rate and mobility of the user equipment. In this paper, we will first perform a comparison in between Reno, NEW RENO, Tahoe and SACK (Selective Acknowledgements) TCP variants in a wireless environment. Number of experiments shows that Tahoe performs better in high segment loss environment. From the experiments results we perform some modifications in the TCP Tahoe congestion control mechanism. Results taken from both modified and unmodified version of TCP Tahoe. For this modification we use two algorithms (1) Slow Start (2) Congestion Avoidance. To solve the congestion in the network we first modify the performance of TCP's sliding window. First of all sender determine the available capacity of network. For congestion control sender keeps two state variables (1) a slow-start/congestion window (*cwnd*) (2) Threshold size (*ssthresh*). These variables are used to change the state from Slow Start to Congestion Avoidance algorithms and from Congestion Avoidance algorithms to slow start. At the beginning of communication session Slow Start provides a control on initial data flow and also during an error recovery. This control is based on acknowledgements received from receiver. Congestion Avoidance algorithm increases the congestion window size (*cwnd*) additively, so that it increases by one segment after each round trip time.

RELATED WORK

Both Slow Start and Congestion avoidance algorithms provide various features in the network like adaptation to network conditions and flow control in error recovery situations. But when an error occurs in the network, these algorithms respond slowly in recovering the network back from data loss to its original throughput. Fast retransmit and Fast Recovery are two different algorithms [1] which help the network to recover with data loss and retrieve to its original throughput. These two algorithms are as follows.

Fast Retransmit

When there is data loss, an immediate acknowledgement is sent to sender. This is called a duplicate acknowledgement. Fast Retransmit means if four acknowledgments are received before the retransmission time out, the sender then immediately retransmit the lost segment. From this point in time every duplicate acknowledgement initiates a new data transmission until an acknowledgement for new data arrives. After that, the *cwnd* is initializes to its initial value and initiate the *Slow Start*.

Fast Recovery

Fast Recovery is performing when duplicate acknowledgements are being received but the network is not totally congested. Thus it is not necessitate to suddenly reducing the flow of data and initiate a *Slow Start* [2]. Hence, when sender receive two duplicate acknowledgements, the *cwnd* is set to half of its previous size and *Congestion Avoidance algorithm* start to work, instead of *Slow Start*.

TCP Variants

- **Reno TCP**

TCP Reno uses both Fast Retransmit and Fast Recovery [3]. If only one segment is lost in this situation TCP RENO performs well. In wireless networks, multiple segments loss performs due to the unreliable media. When number of segment losses performs in network, the *cwnd* reduces its size to half of its previous size for each segment loss. Only Congestion Avoidance algorithm is functioning and due to that recovery it is delayed without the using benefit of Slow Start.

- **Tahoe TCP**

TCP Reno uses much slower Congestion Avoidance mechanism [3] by this a number of segments are lost. Tahoe TCP does not use fast recovery because multiple segments are lost same as Reno. So that in TCP Tahoe used the fast retransmit**.**

- **Sack TCP**

TCP (SACK) also takes the advantage of both Fast Retransmit and Fast Recovery. It also affect from multiple losses. The benefit of TCP SACK is that the receiver sends information to the transmitter for successfully received segments. This retransmission performs only for unacknowledged segments. So the number of retransmitted segments is significantly reduced in the network [3].

- **TCP New Reno**

TCP New Reno [3] includes a small change to the TCP Reno algorithm at the source side that eliminates the waiting time for a retransmit timer to expire when numbers of packets are lost from a window. When a partial acknowledgement is received that acknowledges some received packets, but not acknowledge all of the packets that were outstanding then at the start of that fast recovery procedure the change incorporate the sender's behavior during fast recovery.

UMTS ARCHITECTURE

If you The most important 3G cellular system is Universal Mobile Telecommunications Systems (UMTS) which uses WCDMA for the air interface. UMTS keeps the concepts and solutions of the GSM network but a new infrastructure is required. The UMTS architecture is illustrated in Figure 1 and is composed of three main domains [5]: User Equipment (UE), Core Network (CN) and UMTS Terrestrial Radio Access Network (UTRAN). The UE is the equivalent of the MS in GSM, with added support for UMTS. The Core Network is based on the GSM/GPRS network upgraded in order to support UMTS operation and services. The UTRAN provides the air interface for the UE, and is the equivalent of BSS in GSM, consisting of two main entities: Node B and Radio Network Controller (RNC). Node B is the equivalent of BTS whereas RNC is the equivalent of BSC. A RNC can control one or more Node Bs [10].

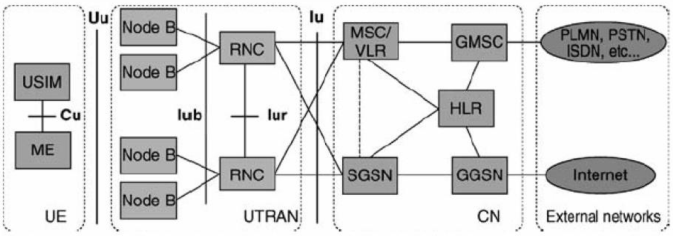

Figure 1: UMTS Architecture

PROPOSED TCP TAHOE MODIFICATIONS

We have done some modifications to the TCP Tahoe variant. We choose Tahoe because we can take advantage of the Fast Retransmit (exponential window opening algorithm) and disabling Fast Recovery which reduces the *cwnd* recovery. The modifications are:

- Before a Fast Retransmit takes place decreasing the number of duplicate acknowledgements required from three to two. This shows the concept of resending a lost segment immediately without delay.
- Increasing the Slow Start initial count from one MSS to four MSS. This will aid a fast recovery and prevent the *cwnd* from loss
- Increasing the receive buffer usage threshold from half the previous *cwnd* size to 0.75. By this the process of slow start continues for a longer period and the *cwnd* comes to its previous size quickly.
- Following algorithm is used to modify TCP Tahoe.

if (Congestion detection || Heavy load){

if (Receive Same ACK 2 Times || Retransmission Timer Timeout) /* Congestion */{

Ssthresh = max (flight size*3/4 , 3*MSS); // Flight size are those data which have no acknowledged

if (Retransmission Timer Timeout)

{*cwnd* = 1; Exit and call slow-start;}// Enter in to slow start phase

else /* Receive Same ACK 2 Time */

cwnd = ssthresh; }// Enter in to congestion avoidance phase

else if (Total drop of packets > 10%)

Go to slow start algorithm}// Enter in to slow start phase

SIMULATION MODEL

OPNET (14.5) simulator is used for deploying UMTS network architecture by using different nodes (mobile & fixed) from object palette. OPNET MODELER [8] is used for design and study communication networks, devices, protocols and applications. It provides a graphical user interface to build simulation models for various network parts from physical layer modulator to application processes [6][7].

- **Simulation Scenario**

For implementation of TCP TAHOE with modified TCP Tahoe five scenarios have been created, i.e TCP-Tahoe, TCP SACK, TCP RENO, TCP NEWRENO and Modified TCP TAHOE . First we analysis the TCP congestion window for all TCP variants. After than we compare the results for all variants of TCP. A modified TCP TAHOE is implemented and results are compared for both TCP TAHOE and modified TCP TAHOE for TCP congestion window. A single scenario completed in all aspects, duplicated and then attributes are set for all the scenarios. Each scenario is employed only for file transfer by using Background class. Each scenario is designed for five users with their movement across Node-B. Along with users, simulation model consists of following entities: one Node-B access points, RNC, SGSN, GGSN and one FTP server for single type of traffic class. For connectivity between nodes various links were used form object palette. After the architecture is completed, the attributes required for each node are defined for network. Applications are defined in the application configuration node and packet discarder utility is used to discard the packet at particular time interval. For each TCP variants different scenarios are designed for measuring the different global and object statistics. Figure 2, Figure 3 and Figure 4 shows the simulation scenario for different TCP variants. Here we show only three scenarios i.e. RENO, SACK and Tahoe.

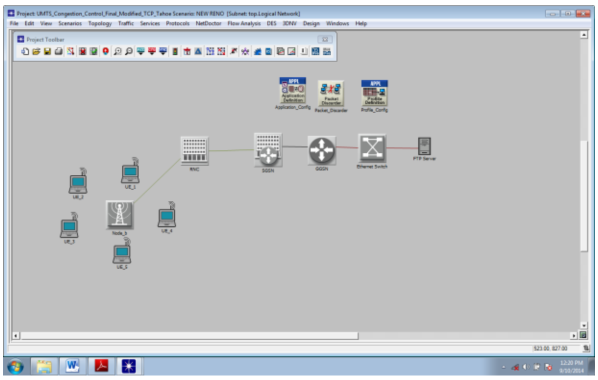

Figure 2: UMTS Simulation Model for TCP Variants (NEW RENO)

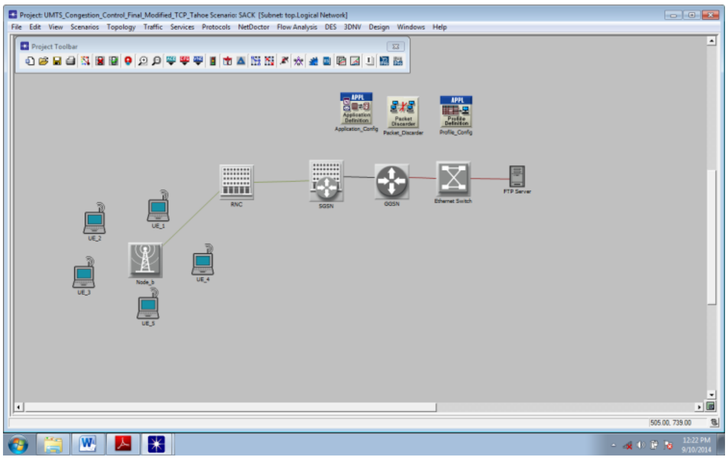

Figure 3: UMTS Simulation Model for TCP Variants (SACK)

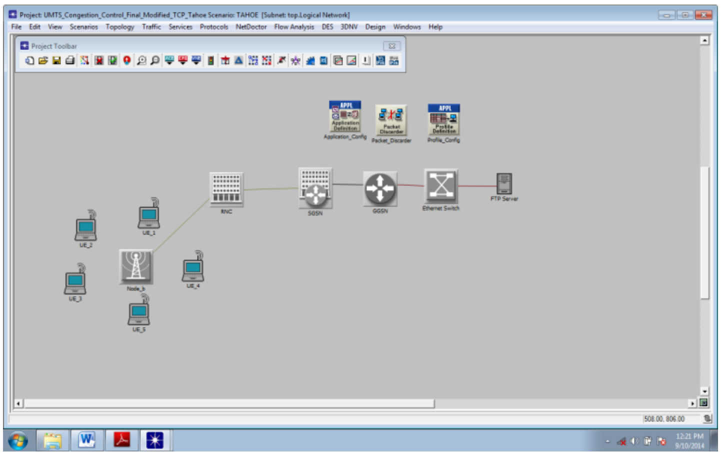

Figure 4: UMTS Simulation Model for TCP Variants (TAHOE)

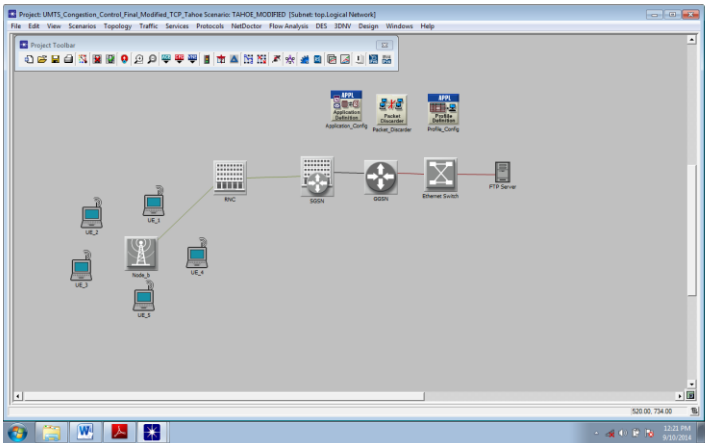

Figure 5: UMTS Simulation Model for TCP Variants (Modified TAHOE)

SIMULATION PARAMETERS

For analysis of results various parameters needs to be considered are Scenario parameters, profile configuration parameters and Packet Discarder parameters.

- **Scenario Parameters:** For each scenario certain parameters are considered and needs to be set as shown in table 2.

Table 1: Simulation Parameter

Simulation Parameters	Value
Simulation Time	600 Sec
Number of Nodes	05
Environment Size	Logical Environment
Traffic Type	Constant Bit Rate
Seed	300
Value per Statistics	300
Update Interval	500000
Simulation	Based on Kernel type Preferences
Number of runs	One for each scenario

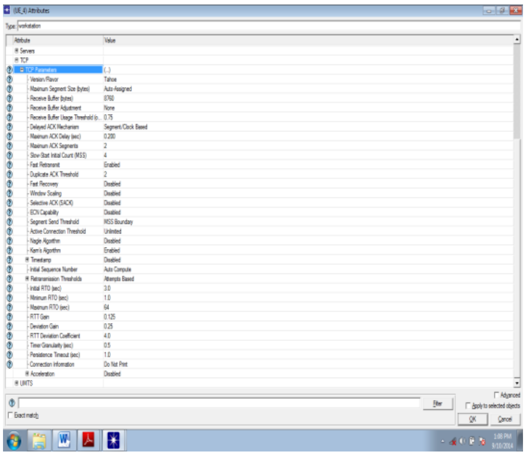

Figure 6: Modified TCP Tahoe Parameters

SIMULATION RESULTS

Analyzing *TCP* Variants results by implementing the changes perform in TCP Tahoe. Number of simulations was executed and results recorded saved and compared. As shown from Figures 10 and 11, modified Tahoe TCP gives a minimum *congestion window recovery time* and a minimum *FTP download response time*.

- **Global Stastics**

These static is collected for FTP download response time. Figure 7 shows that before modification FTP download response time for TCP New RENO is greater than other variants

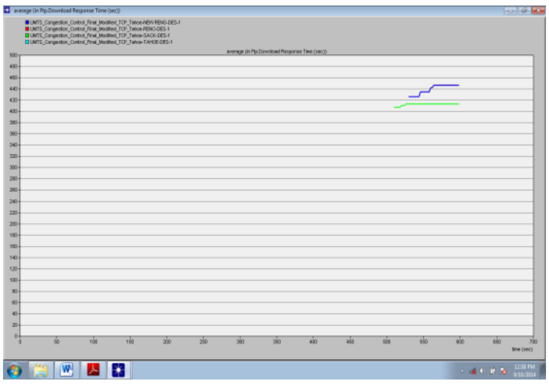

Figure 7: FTP Download Response Time for TCP Variants

- **Object Statics**

Object statics is collected for various network objects i.e. for FTP server and for various user equipment. For each node and user equipment TCP congestion window is shown in results. Figure 8 and Figure 9 shows the results for unmodified TCP Tahoe and Figure 10 and Figure 11 shows the TCP Congestion window results for modified TCP Tahoe.

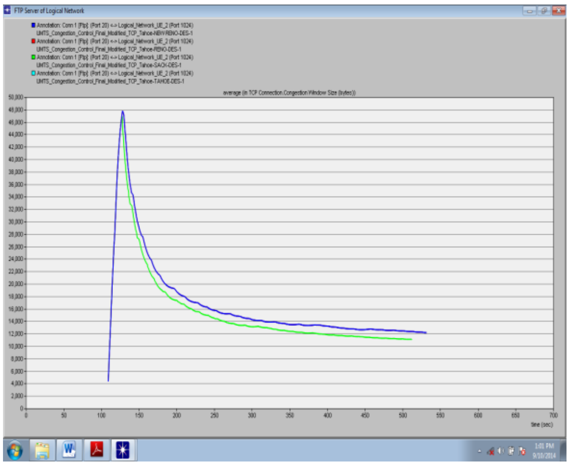

Figure 8: FTP Download Response Time for TCP Variants

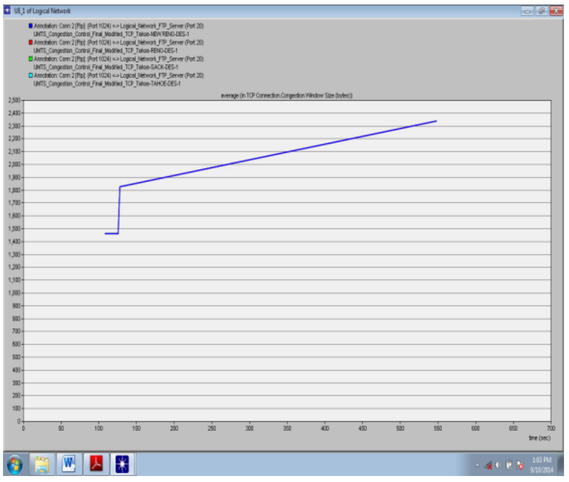

Figure 9: TCP Congestion Window Size at UE_1 with TCP Tahoe

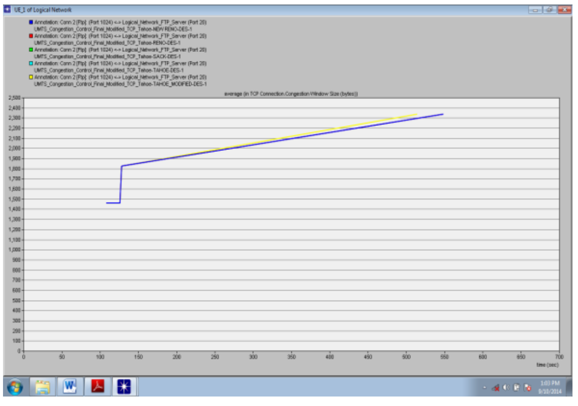

Figure 10: TCP Congestion Window Size at UE_1 with Modified TCP Tahoe

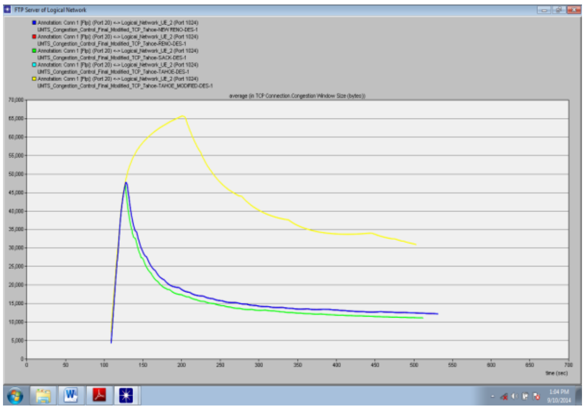

Figure 11: TCP Congestion Window Size at FTP Server with Modified TCP Tahoe

CONCLUSIONS

In this research paper a comparison is perform for different TCP variants. The results show that when unmodified version of TCP Tahoe compared to other variants NEW RENO performs better rather than TCP Tahoe. When modified TCP Tahoe is used for comparisons with other variants modified TCP Tahoe performs better on wireless links. Figure 10 & Figure 11 shows the results for modified Tahoe and it represents that TCP Tahoe works best under a wireless scenario.

ACKNOWLEDGEMENTS

The Authors are grateful to Govt. Engineering College TEQIP_II Project for funding and providing support to carry out this research work.

REFERENCES

1. W. Stevens, "*TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms*". Network Working Group, RFC: 2001, January 1997.
2. G. Corral, A. Zaballos, "*Simulation-based study of TCP flow control mechanisms using OPNET Modeler"*.
3. K. Fall, S. Floyd, *"Simulation Based Comparisons of Tahoe, Reno and SACK TCP"*1995.
4. Andrew S. Tanenbaum, *"Computer Networks"* The Transport Layer, 3rd Edition.
5. 3GPP, "3G TS 25.401: UTRAN Overall Description (Release 1999),"
6. S.-J. Seok, S.-K. Youm, S.-W. kim, and C.-H. Kang, "*A modification of TCP flow control for improving end-to-end TCP performance over networks with wireless links"* Computer Communications, vol. 26, pp. 1998-2010, 2003.
7. OPNET Modeler 14.5, UMTS model user guide, OPNET documentation.
8. OPNET for research, www.riverbed.com
9. http://www.opnet.com
10. B. Walke, R. Seidenberg and M. P. Althoff, "*UMTS: The Fundametals*",ISBN: 0-470-84557-0, John Wiley & Sons Ltd, 2003.